IN THE CLAIMS:

Please cancel claim 37.

Please amend the claims as follows:

1. (CURRENTLY AMENDED) Apparatus for enabling an instruction to control data flow bypassing hardware within a processor of a programmable processing engine, the apparatus comprising:

a pipeline of the processor, the pipeline having a plurality of stages including instruction decode, writeback, and execution stages, the execution stage having a plurality of parallel execution units; and

an instruction set of the processor, the instruction set defining a first register decode value~~, that specifies one of a first register decode value which~~ that defines source operand bypassing that allows source operand data to be shared among the plurality of execution units, and a second register decode value that defines result bypassing that allows bypassing of a result from a previous instruction executing in pipeline stages of the processor.

2. (ORIGINAL) The apparatus of Claim 1 further comprising:

a register file containing a plurality of general-purpose registers for storing intermediate result data processed by the execution units; and

a memory for storing one of transient data unique to a specific process and pointers referencing data structures.

3. (CURRENTLY AMENDED) The apparatus of Claim 1 wherein the second register decode value comprises:

one of a result bypass (RRB) operand and an inter-unit result bypass (RIRB) operand, each of which explicitly controls data flow within the pipeline of the processor.

F1

ET

4. (ORIGINAL) The apparatus of Claim 3 wherein the execution units comprise a current execution unit and an alternate execution unit, and wherein the RRB operand denotes the current execution unit and the RIRB operand denotes the alternate execution unit.

5. (ORIGINAL) The apparatus of Claim 3 wherein the RRB operand explicity infers feedback of the data delivered from a current one of the execution units to an input register of the current execution unit over a feedback path.

6. (ORIGINAL) The apparatus of Claim 5 wherein the writeback stage comprises an interstage register and wherein the RRB operand enables bypassing write-back of the data processed by the execution units to one of the register file or the interstage register of the writeback stage.

7. (CURRENTLY AMENDED) The apparatus of Claim 2 wherein the first register decode value comprises a source bypass (RISB) operand that allows source operand data to be shared among the parallel execution units of the pipelined processor.

8. (ORIGINAL) The apparatus of Claim 7 wherein the execution units comprise a main execution unit and a secondary execution unit, and wherein the RISB operand allows the secondary execution unit to receive data stored at an effective memory address specified by a displacement operand in the previous instruction executed by the main execution unit.

9. (CURRENTLY AMENDED) A method for enabling an instruction to control data flow bypassing hardware within a pipelined processor of a programmable processing engine, the method comprising the steps of:

defining a first register decode value that ~~specifies one of a first register decode value which~~ defines source operand bypassing of source operand data and a second register decode value that defines result bypassing of a result from a previous instruction executing in pipeline stages of the processor; and

identifying a pipeline stage register for use as a source operand in an instruction containing the first or the second register decode value.

10. (CURRENTLY AMENDED) The method of Claim 9 further comprising: ~~the step of~~

explicitly controlling data flow within the pipeline stages of the processor through the use of a register result bypass (RRB) operand in said second register decode value.

11. (CURRENTLY PRESENTED) The method of Claim 10 further comprising:

including pipeline stages having instruction decode, writeback and execution stages, and wherein the execution stage has multiple parallel execution units including a current execution unit and an alternate execution unit.

12. (CURRENTLY AMENEDED) The method of Claim 11 wherein the step of explicitly controlling comprises ~~the steps of~~:

retrieving data from the current execution unit; and

returning the data to an input execution register specified by the RRB operand, thereby bypassing write-back of the data to either a register file or memory at the writeback stage.

13. (CURRENTLY AMENEDED) The method of Claim 12 wherein the step of identifying further comprises ~~the steps of~~:

explicitly specifying the pipeline stage register to be used as the source operand for the instruction.

14. (PREVIOUSLY PRESENTED) The method of Claim 13 further comprising:

encoding the RRB operand in fewer bits than a regular register operand.

15. (PREVIOUSLY PRESENTED) The method of Claim 9 further comprising:

including pipeline stages having instruction decode, writeback and execution stages, and wherein the execution stage has multiple parallel execution units including a current execution unit and an alternate unit; and

sharing source operand data among the parallel execution units of the pipelined processor through the use of a source bypass (RISB) operand in said first register decode value.

16. (CURRENTLY AMENEDED) The method of Claim 15 wherein the step of sharing further comprises: ~~the step of~~

receiving data at the alternate execution unit, the data stored at a memory address specified by a displacement operand in a previous instruction executed by the current execution unit of the processor.

17. (CURRENTLY AMENEDED) The method of Claim 16 wherein the step of sharing further comprises: ~~the step of~~

realizing two memory references through the use of a single bus operation over a local bus.

18. (CURRENTLY AMENEDED) The method of Claim 17 wherein the step of sharing further comprises: ~~the step of~~

encoding the RISB operand with substantially fewer bits than those needed for a displacement address.

19. (PREVIOUSLY PRESENTED) A computer readable medium containing executable program instructions for enabling an instruction to control data flow bypassing hardware within a pipelined processor of a programmable processing engine, the executable program instructions comprising program instructions for:

defining a first register decode value that ~~specifies one of a first register decode value that~~ defines source operand bypassing of source operand data and a second register decode value that defines result bypassing of a result from a previous instruction executing in pipeline stages of the processor; and

identifying a pipeline stage register for use as a source operand in a current instruction containing the register decode value.

20. (ORIGINAL) The computer readable medium of Claim 19 further comprising program instructions for explicitly controlling data flow within the pipeline stages of the processor through use of a register result bypass operand.

21. (ORIGINAL) The computer readable medium of Claim 20 further comprising program instructions for sharing source operand data among parallel execution units of the pipelined processor through the use of a source bypass operand.

22. (CANCELLED)

23. (CANCELLED)

24. (CANCELLED)

25. (CANCELLED)

26. (CANCELLED)

27. (CANCELLED)

28. (PREVIOUSLY PRESENTED) A processor comprising:

a first execution unit having at least one first input and a first output;

at least one second execution unit having at least one second input and a second output;

F1

a first input register connected to said at least one first input;

a second input register;

a multiplexer having a first input from said first input register, a second input from said second input register, and an output to said at least one second execution unit; and

a register decode value that specifies bypassing data from said first input register to said at least one second execution unit via said multiplexer.

29. (PREVIOUSLY PRESENTED) The processor of claim 28 further comprising:

a first instruction having at least one first source operand and a first destination, said first execution unit processing said first instruction;

a second instruction having at least one second source operand and a second destination operand, said at least one second source operand is the same as said at least one first source operand; and

means for replacing said at least one second source operand with said register decode value.

30. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

a register file connected to said first input register and said second input register; and

means for loading said at least one first and said at least one second source operands from said register file.

31. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

a memory connected to said first input register; and

means for loading said at least one first and said at least one second source operands from said memory.

32. (PREVIOUSLY PRESENTED) The processor of claim 29, said means for replacing further comprising:

an instruction decode mechanism; and

means for said multiplexer choosing input from said first input register.

33. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

said register decode value having fewer bits than said at least one second source operand.

34. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

a displacement value within said at least one first and said at least one second source operands, said displacement value specifying an effective memory address where data is stored.

35. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

a displacement value within said first destination operand, said displacement value specifying an effective memory address where data is stored.

36. (PREVIOUSLY PRESENTED) Electromagnetic signals propagating over a computer network comprising:

said electromagnetic signals carrying instruction for execution on a processor for performing the method of claim 9.

37. (CANCELLED)

38. (PREVIOUSLY PRESENTED) The method of Claim 9 further comprising:

including pipeline stages having instruction decode, writeback and execution stages, and wherein the execution stage has multiple parallel execution units including a current execution unit and an alternate execution unit; and

explicitly controlling data flow within the pipeline stages of the processor through the use of a register result bypass (RIRB) operand to bypass the writeback stage and to allow result data from an alternate execution unit to flow directly to an input execution register.

39. (PREVIOUSLY PRESENTED) The apparatus of Claim 3 wherein the RIRB operand explicitly infers feedback of the data delivered from an alternate one of the execution units to an input register of the current execution unit over a feedback path.

40. (PREVIOUSLY PRESENTED) Apparatus for enabling an instruction to control data flow within a processor of a programmable processing engine, the apparatus comprising:

a pipeline of the processor, the pipeline having a plurality of stages including instruction decode, writeback and execution stages, the execution stage having a plurality of parallel execution units;

a multiplexer connecting parallel execution units; and

an instruction set of the processor, the instruction set defining a register decode value that controls said multiplexer to bypass a source operand from a previous instruction executing in pipeline stages of the processor to the source operand of a current instruction.

41. (PREVIOUSLY PRESENTED) The apparatus of Claim 40 further comprising:

a register file containing a plurality of general-purpose registers for storing intermediate result data processed by the execution units.

42. (PREVIOUSLY PRESENTED) The apparatus of Claim 40 further comprising:

a memory for storing one of transient data unique to a specific process and pointers referencing data structures.

43. (PREVIOUSLY PRESENTED) The apparatus of Claim 40 wherein the register decode value comprises:

a source bypass operand (RISB) that allows source operand data to be shared among the parallel execution units of the pipelined processor.

ET
A

44. (PREVIOUSLY PRESENTED) The apparatus of Claim 40 wherein the execution units comprise:

a main execution unit and a secondary execution unit, wherein the RISB operand allows the second execution unit to receive data stored at a memory address specified by a displacement operand in the previous instruction executed by the main execution unit.

45. (PREVIOUSLY PRESENTED) The apparatus of Claim 44 wherein the instruction set of the processor comprises:

an opcode directed to the main execution unit, said opcode having sufficient bits to encode a displacement operand;

an opcode directed to the secondary execution unit; and

micro-opcodes to initiate memory prefetches without requiring a dedicated instruction.

46. (PREVIOUSLY PRESENTED) A method for enabling an instruction to control data flow within a pipelined processor of a programmable processing engine, the method comprising the steps of:

defining a register decode value that specifies one of source operand bypassing from a previous instruction executing in pipeline stages of the processor; and

identifying a pipeline stage register for use as a source operand in an instruction containing the register decode value.

47. (PREVIOUSLY PRESENTED) The method of Claim 46 further comprising:

including pipeline stages having instruction decode, writeback and execution stages, and wherein the execution stage has multiple parallel execution units including a current execution unit and an alternate execution unit.

48. (PREVIOUSLY PRESENTED) The method of claim 47 further comprising:

sharing source operand data among the parallel execution units of the pipelined processor through the use of a source bypass (RISB) operand.

49. (PREVIOUSLY PRESENTED) The method of claim 48 further comprising:

receiving data at said alternate execution unit, the data stored at a memory address specified by a displacement operand in a previous instruction executed by said current execution unit of the processor.

50. (PREVIOUSLY PRESENTED) The method of claim 49 further comprising:

realizing two memory references through the use of a single bus operation over a local bus.

51. (PREVIOUSLY PRESENTED) The method of claim 49 further comprising:

encoding the RISB operand with substantially fewer bits that those needed for a displacement address.

52. (CURRENTLY AMENDED) Electromagnetic signals propagating on a computer network, comprising:

said electromagnetic signals carrying instruction for the practice of the method of ~~Claim 9 or~~ Claim 46.

53. (CURRENTLY AMENDED) A computer readable media comprising:

said computer readable media containing executable program instruction for the practice of the method of ~~Claim 9 or~~ Claim 46.